What is claimed is:

1. A cache, comprising:

a front-end interface that receives data access requests that specify respective data storage addresses;

a back-end interface that can retrieve data identified by the data storage addresses;

cache storage formed by at least two disks; and

a cache manager that services at least some of the requests received at the front-end interface using data stored in the cache storage.

2. The cache of claim 1, wherein the front-end interface comprises an interface conforming to a protocol.

3. The cache of claim 2, wherein the protocol comprises at least one of the following: SCSI (Small Computer System Interface), Fibre Channel, Infiniband, and IDE (Integrated Device Electronics).

4. The cache of claim 1, wherein the disks comprise disks having platters less than 3.5 inches in diameter.

5. The cache of claim 4, wherein the disks comprise disks having at least one of the following platter sizes: 2.5 inches, 1.8 inches, and 1 inch in diameter.

6. The cache of claim 1, wherein the cache implements a RAID (Redundant Array of Independent Disks) scheme using the disks.

7. The cache of claim 1, wherein the cache performs at least one of the following operations: requesting data from a back-end storage system, retrieving requested data from the disks, sending data to the back-end system for writing, determining the location of back-end system data within the disks, and removing data from the disks.

8. The cache of claim 1, wherein the addresses specify storage locations of a back-end storage system that includes a collection of one or more disks.

9. The cache of claim 1, wherein the requests comprise I/O (Input/Output) requests.

10. The cache of claim 1, wherein the data storage addresses comprise data storage addresses within an address space.

11. The cache of claim 10, wherein the address space comprises an address space of back-end storage.

12. The cache of claim 10, wherein the address space comprises an address space of a different cache.

13. The cache of claim 1, wherein the cache storage comprises cache storage having more than one disk spindle.

14. A method of servicing data access requests at a cache, the method comprising:

receiving the data access requests at the cache, the cache having cache storage formed by at least two disks, the requests specifying respective data storage addresses; and

servicing at least some of the requests using data stored in the disks.

15. The method of claim 14, wherein the requests comprise requests conforming to a protocol.

16. The method of claim 15, wherein the protocol comprises at least one of the following: SCSI, Fibre Channel, Infiniband, and IDE.

17. The method of claim 14, wherein the requests comprise at least one read request.

18. The method of claim 14, wherein servicing the requests comprises retrieving data from the back-end storage and storing the data in at least one of the disks.

19. The method of claim 18, wherein storing the data comprises storing the data in accordance with a RAID scheme.

20. The method of claim 14, wherein servicing the requests comprises determining whether the collection of disks currently stores the requested data.

21. The method of claim 14, wherein the data storage addresses comprise data storage addresses within an address space.

22. The method of claim 21, wherein the address space comprises an address space of a back-end storage system formed by a collection of disks.

23. A data storage system, comprising:

a back-end storage system having an address space, addresses in the address space identifying blocks of storage; and

a cache for the back-end storage system having a lesser storage capacity than the back-end storage system, the cache including:

a front-end interface that receives I/O (Input/Output) requests that specify respective addresses of back-end storage blocks;

a back-end interface that communicates with the back-end storage system;

cache storage formed by at least two disks having platter diameters less than 3.5 inches; and

a cache manager that services at least some of the I/O requests received via the front-end interface using blocks temporarily stored in the cache storage.